

RADIO LINK PROTOCOL ENHANCEMENTS FOR DYNAMIC CAPACITY WIRELESS DATA CHANNELS

BACKGROUND OF THE INVENTION

5

I. Field of the Invention

The present application is a continuation-in-part application of
copending U.S. Patent Application Serial No. 09/321,296, filed May 27, 1999,
entitled " RADIO LINK PROTOCOL ENHANCEMENTS FOR DYNAMIC
10 CAPACITY WIRELESS DATA CHANNELS" and as such relates to wireless
communications. More particularly, the present invention relates to an
improved method and system for reliably transmitting data through a wireless
channel while minimizing the overhead inherent in the error control protocol.

U.S. PAT
6,507,582

15 II. Description of the Related Art

The use of code division multiple access (CDMA) modulation techniques
is one of several techniques for facilitating communications in which a large
number of system users are present. Other multiple access communication
20 system techniques, such as time division multiple access (TDMA), frequency
division multiple access (FDMA) and AM modulation schemes such as
amplitude companded single sideband (ACSSB) are known in the art. These
techniques have been standardized to facilitate interoperation between
equipment manufactured by different companies. Code division multiple access
25 communications systems have been standardized in the United States in
Telecommunications Industry Association TIA/EIA/IS-95-B, entitled "MOBILE
STATION-BASE STATION COMPATIBILITY STANDARD FOR DUAL-MODE
WIDEBAND SPREAD SPECTRUM CELLULAR SYSTEMS", incorporated by
reference herein, and hereinafter referred to as IS-95.

30 IS-95 was originally optimized for transmission of variable-rate voice
frames. In order to support two-way voice communications, as typified in
wireless phone applications, it is desirable that a communication system
provide fairly constant and minimal data delay. For this reason, IS-95 systems
are designed with powerful forward error correction (FEC) protocols and
35 vocoders which are designed to respond gracefully to voice frame errors. Error
control protocols which require frame retransmission procedures add
unacceptable delays to voice transmission, so are not designed into the IS-95
specification.

1c648 U.S. PRO
07/28/99

The optimizations which make the standalone IS-95 specification ideal for voice applications make it difficult to use for packet data applications. In many non-voice applications, such as the transmission of Internet protocol (IP) data, the delay requirements of the communication system are much less stringent than in voice applications. In the Transmission Control Protocol (TCP), probably the most prevalent of protocols used in an IP network, virtually infinite transmission delays are allowed in order to guarantee error-free transmission. TCP uses retransmissions of IP datagrams, as IP packets are commonly called, to provide this transport reliability.

IP datagrams are generally too large to fit into a single IS-95 frame. Even after dividing an IP datagram into segments small enough to fit into a series of IS-95 frames, an entire series of IS-95 frames would have to be received without error for the single IP datagram to be useful to TCP. The frame error rate typical of an IS-95 system make the probability of error-free reception of all segments of a single datagram very low.

As described in IS-95, alternative service options enable the transmission of other types of data may in lieu of voice frames. The TIA/EIA/IS-707-A, entitled "DATA SERVICE OPTIONS FOR SPREAD SPECTRUM SYSTEMS", hereafter referred to as IS-707, describes procedures used in the transmission of packet data in an IS-95 system.

Radio Link Protocol (RLP) is described in TIA/EIA/IS-707-A.8, entitled "DATA SERVICE OPTIONS FOR SPREAD SPECTRUM SYSTEMS: RADIO LINK PROTOCOL TYPE 2", hereinafter referred to as RLP2, and incorporated herein by reference. RLP2 incorporates an error control protocol with frame retransmission procedures over the IS-95 frame layer. RLP is of a class of error control protocols known NAK-based ARQ protocols, which are well known in the art. The IS-707 RLP, facilitates the transmission of a byte-stream, rather than a series of voice frames, through an IS-95 communication system.

Several protocol layers typically reside above the RLP layer. IP datagrams, for example, are typically converted into a Point-To-Point Protocol (PPP) byte stream before being presented as a byte stream to the RLP protocol layer. As the RLP layer ignores the protocol and framing of higher protocol layers, the stream of data transported by RLP is said to be a "featureless byte stream".

RLP was originally designed to satisfy the requirements of sending large frames through an IS-95 channel. For example, if an IP datagram of 500 bytes were to be simply sent in IS-95 frames carrying 20 bytes each, the IP datagram would fill 25 consecutive IS-95 frames. Without some kind of error control layer, all 25 of these frames would have to be received without error in order

for the IP datagram to be useful to higher protocol layers. On an IS-95 channel having a 1% frame error rate, the effective error rate of the IP datagram delivery would be $(1 - (0.99)^{25})$, or 22%. This is a very high error rate compared to most networks used to carry Internet Protocol traffic. RLP was designed as a link
5 layer protocol which would decrease the error rate of IP traffic to be comparable to the error rate typical of a 10Base2 ethernet channel.

The International Telecommunications Union recently requested the submission of proposed methods for providing high rate data and high-quality speech services over wireless communication channels. A first of these
10 proposals was issued by the Telecommunications Industry Association, entitled "The cdma2000 ITU-R RTT Candidate Submission", and hereinafter referred to as cdma2000. A second of these proposals was issued by the European Telecommunications Standards Institute (ETSI), entitled "The ETSI UMTS Terrestrial Radio Access (UTRA) ITU-R RTT Candidate Submission", also
15 known as "wideband CDMA" and hereinafter referred to as W-CDMA. A third proposal was submitted by U.S. TG 8/1 entitled "The UWC-136 Candidate Submission", hereinafter referred to as EDGE. The contents of these submissions is public record and is well known in the art.

RLP2 is optimized for use with IS-95B, in which the rate set, derived
20 from channel capacity, used during a packet data call remains essentially fixed for the duration of the call. Based on this assumption of fixed rate sets, RLP2 is designed with the assumption that retransmitted RLP frames can be sent within a maximum of three consecutive RLP segments. The probability that one of three segments is lost, causing the loss of a retransmitted RLP frame, has been
25 deemed acceptable by the designers of RLP2.

In cdma2000, however, the channel capacity available to a single user, and hence the maximum data rate used during a packet data call, can vary widely and rapidly. For example, during the course of a single cdma2000 call, the supplemental channel capacity used by a packet data service option may
30 vary from 9.6 kilo-bits per second (kbps) to more than 307kbps. In a simple extension of the RLP2, the maximum number of segments used during retransmissions could be increased as necessary to accommodate the change in data rates. Because the capacity of a call channel may diminish during a call, a full-rate frame transmitted unsuccessfully at a high rate might span 30 or more
35 consecutive lower-rate-set cdma2000 segments. The high likelihood that one or more of those cdma2000 segments makes such a simple extension of the RLP2 largely impractical for use with cdma2000.

RLP2 has been optimized to require minimal protocol overhead space over the two IS-95 rate sets, known as Rate Set 1 (RS1) and Rate Set 2 (RS2). The

RLP frame sequence numbers in RLP2 are 8 bits long, a size which is ideal for computer processing. Since the rate sets specified in cdma2000 include RS1 and RS2, it would be highly desirable for an RLP designed for cdma2000 (RLP2000) to be at least as efficient as RLP2 when used at RS1 and RS2. Because switching
5 RLP protocols whenever the rate set changes would add complexity to RLP2000, it is desirable that a single RLP2000 protocol efficiently support RS1, RS2, and all the higher cdma2000 data rates without requiring resynchronization or substantial protocol complexity.

Another problem that arises in using RLP2 on a high-data-rate channel is
10 transmit sequence number ambiguity. Though the sender and receiver in an RLP2 link maintain 12-bit RLP sequence numbers, all data-bearing frames carry only the least-significant 8 bits of their sequence numbers. The number of sequence-number bits contained in over-the-air RLP frames determines the number of outstanding frames that can be unambiguously tracked by the
15 receiver of the data. The sequence number space within such a block of outstanding frames is often referred to in the art as a "sliding window." In RLP2, each 8-bit over-the-air sequence number can be used to unambiguously identify one 12-bit sequence number, and hence one data frame, within a window of 256 consecutive sequence numbers. Any frame whose 12-bit
20 sequence number falls outside the sliding window will have a counterpart within the sliding window whose 12-bit sequence number has the same least-significant 8 bits. The receiver of both of these data frames will be unable to tell the frames apart from each other.

In older versions of the IS-95 protocol (earlier than IS-95B), one data
25 frame is transmitted every 20 milliseconds (50 per second), and the 8-bit RLP sequence number space spans approximately 5 seconds of consecutive data frames. On such a channel, a deep fade lasting longer than 5 seconds could cause the loss of more than 255 consecutive data frames. As a result, the 8-bit RLP sequence numbers would "wrap-around" and can cause ambiguity for
30 new-data frames and for retransmit frames containing 8-bit shortened sequence numbers. With older IS-95 implementations, the wrap-around problem does not occur, because a 5-second fade causes the call to drop entirely. Whenever RLP performs a reset, data in the RLP byte stream is lost.

Newer protocols such as IS-95B, however, allow sending multiple frames
35 in a single 20-millisecond period. This makes it possible to exhaust the over-the-air sequence numbers in a much shorter fade. For example, in IS-95B, which allows sending eight frames per 20-millisecond period, an 8-bit sequence number window can be exhausted in 640 milliseconds. In RLP2, which is designed for use with IS-95B, RLP performs a reset whenever 255 consecutive

received frames are classified as erasures. Thus, a deep fade on an RLP2 link which lasts as little as 640 milliseconds may cause an RLP reset, along with its associated data loss.

As discussed in the aforementioned RLP2 and all existing RLP implementations, three variables are maintained at either side of an RLP protocol link. These variables are V(R), V(N) and V(S). As discussed in the RLP standards, V(R) contains the expected value of the RLP frame sequence number field in the next new traffic channel frame to be received. V(N) contains the sequence number of the next needed traffic channel frame not received in sequence. The sequence number field 'SEQ' in each new RLP data frame sent and in each RLP idle frame sent shall be set to V(S). Each of the variables V(R), V(N) and V(S) are the shortened (8-bit), over-the-air versions of the full (12-bit) sequence numbers L_V(R), L_V(N) and L_V(S) also maintained at either side of an RLP protocol link.

RLP is a NAK-based protocol, which means that a sender continues to send new data until it receives a NAK. Since a deep fade often causes the loss of packets in both directions of a channel, a deep fade can cause the loss of both data frames sent in one direction and the corresponding NAK frames in the other direction. The same problem can occur whether frame sequence numbers or byte sequence numbers are used, because a minimal number of bits are used in over-the-air sequence numbers in either method. A deep fade that causes the loss of a large number of sequential frames can cause sequence number wrap-around with either sequence number scheme.

Thus, a method of resolving sequence number ambiguity that is simple, efficient and does not significantly add to over-the-air sequence number overhead is highly desirable.

SUMMARY OF THE INVENTION

The present invention may be used to design an enhanced RLP to enable efficient transmission of a featureless byte stream through a channel of varying capacity. An exemplary embodiment of the invention is as efficient as RLP2 when used over a channel having the same capacity as IS-95 RS1 and RS2. At the same time, an enhanced RLP, designed in accordance with the current invention, also enables efficient transmission of data at varying channel capacities up to and exceeding the maximum specified in cdma2000. The present invention is applicable to any communication system employing transmission of a byte stream over a wireless channel. The present invention is applicable to systems such as cdma2000, W-CDMA, and EDGE, wherein a byte

stream may be carried within over-the-air frames specified for use by the wireless communication system.

The efficiency of the embodiments of the invention at varying rates is made possible by changing the interpretation of the sequence numbers carried in the RLP protocol header. In RLP2, sequence numbers are used to denote frame numbers. This is appropriate for RLP2, as the channel capacity used in a packet data call, and hence the maximum number of data bytes carried in a full-rate frame, are both constant. RLP2 uses a one-byte frame sequence number, and frames are transmitted at 20 millisecond (ms) intervals. When an RLP2 frame is lost during transmission, the data of the lost frame is segmented into as many as three retransmit segments, each having the same sequence number as the original, lost frame.

In trying to adapt RLP2 for use over channels with widely varying capacity, difficulty arises when a frame transmitted on a high-capacity channel (for example, 307kbps) must be retransmitted on a low-capacity channel (for example, 9.6kbps). Using a frame interval of 20ms, a full-rate frame on a 307kbps channel could carry as many as 750 bytes of data. Such a frame could be lost during transmission, and at the same time, the channel capacity might be reduced to 9.6kbps. In RLP2, the capacity of a 9.6kbps full-rate 20ms frame is 20 bytes. In a simple extension of the maximum allowable retransmission segments, successful retransmission of a single frame of 750 bytes of data would require successful transmission of approximately 38 consecutive 9.6kbps full-rate RLP2 retransmit segments. Because all retransmit segments would have the same sequence number, the loss of one of those 38 retransmit segments would cause the loss of the entire retransmitted frame. The receiver could not negatively-acknowledge (NAK) individual retransmit segments. If the over-the-air frame error rate were 1%, the probability of successful transmission of 38 consecutive full-rate RLP2 retransmit segments would be approximately 68%. In this scenario, the retransmission of the segments would often fail, causing data loss and a break in the byte stream due to RLP2 resynchronization. Thus, such a simple extension of RLP2 would frequently result in lost data whenever a high-capacity full-rate frame had to be retransmitted over a low-capacity channel.

One way to accomplish more reliable data retransmission would be to use a byte sequence number in the RLP header instead of a frame sequence number. Then, upon the loss of a large, high-rate RLP frame followed closely by a decrease in channel capacity, the data in the lost frame could be divided into small, independent RLP retransmit frames. The receiver would not be required to receive 38 consecutive retransmit frames without error. The

receiver could accept whichever retransmit frames it successfully received and simply negatively acknowledge (NAK) any lost retransmitted frames. Again using an over-the-air frame error rate were 1%, the probability of the same retransmit segment being lost twice in a row would be 0.01%.

5 One disadvantage of using a byte sequence number instead of a frame sequence number is the larger number of bits in a byte sequence number to represent the same data. If a byte sequence number were used in a 9.6kbps full-rate RLP2 frame, the sequence number would have to be 5 bits longer than the 8-bit frame sequence number. In cdma2000, where channel capacity may vary
10 from 9.6kbps to 32 times that capacity (approximately 307kbps), a full-rate 307kbps frame could carry as many as 750 RLP data bytes. The number of byte sequence number bits necessary to track the same duration of 20-millisecond frames as in RLP2 is at least 18 bits. To make room for an 18-bit byte sequence number in a 9.6kbps RLP frame, the frame would be able to carry two fewer
15 data bytes, a decrease of 10%.

 The embodiments of the invention provide the benefits of large byte sequence numbers while transmitting a fraction of the sequence number bits in the majority of over-the-air frames. In one embodiment of the invention, a 20-bit byte sequence number is used to track received data. The number of bytes
20 which may be tracked with a sequence number is called the sequence number space. In the case of a 20-bit sequence number, the size of the sequence number space is 2^{20} .

 Gaining the benefits of a large sequence number without adding to the average frame header size is accomplished by carefully selecting portions of the
25 sequence number space which will go unassigned to transmitted data bytes. In other words, some of the sequence number space will not be used to track actual transmitted bytes, and may be viewed as wasted. The size of the sequence number chosen is chosen such that wastage of the sequence number space is permissible without impacting the performance of the protocol. For
30 example, if an 18-bit sequence number is necessary to endure a 5-second transmission loss on a 307kbps channel, the use of a 20-bit sequence number allows three fourths of the sequence number space to go unused without impacting the maximum allowable length of transmission loss.

 The unused portion of sequence number space is chosen such that the
35 first byte of each new transmitted data frame starts at a predetermined distance, called a page size, from the first byte of the previous data frame. For example, if the first byte in frame n has a sequence number of 1000, and the page size is 100, the first byte of frame $n+1$ will start on the next page with a sequence number of 1100. If frame n only carries 40 bytes, numbered 1000 to 1039, then

the sequence number space from 1040 to 1099 would go unused. The motivation to allow the seeming waste of sequence number space allows a decrease in the bit size of the sequence number sent in the frame. In the example just shown, the sequence number could be divided by 100 before
5 inserting it into the frame, and could therefore be represented by at least 6 fewer bits. In a preferred embodiment of the invention, page sizes are powers of 2, such as 64, to facilitate computer software manipulation of sequence numbers.

In the previously described scenario, in which a 750 data bytes in a high-
10 rate frame are lost, the preferred embodiment of the invention easily adapts to retransmission on high capacity and low capacity channels alike. In a high capacity channel, the data is easily retransmitted in one or two retransmit frames. If, however, the capacity of the channel has decreased, the data bytes to be retransmitted are divided among several independent retransmit frames,
15 each having its own sequence number. The use of independent retransmit sequence numbers has advantages over the segmented retransmission specified by RLP2. If a single RLP2 retransmit segment is lost in transmission, then all the segments carrying the same sequence number must be again retransmitted in order to recover the data of the lost segment. In contrast, if one or more of the
20 independently-numbered retransmit frames are lost in transmission, the receiver can negatively-acknowledge (NAK) the individual, lost retransmit frame. After receiving the second NAK, the transmitter may send the individual retransmit frame for a second time. Again using an over-the-air frame error rate of 1%, the probability of the same retransmit frame being lost
25 twice in a row is 0.01%. RLP resynchronization, and its associated data loss and byte stream discontinuity, is rarely necessary.

One disadvantage of using sequence numbers corresponding to bytes instead of frames is that more bits are generally needed to represent the sequence number. This can cause a slight increase in the number of
30 independently-numbered retransmit frames needed to carry the data of a lost frame. In an exemplary embodiment of the invention, however, this impact is minimized by sometimes omitting most significant and least significant bits from the sequence number.

Embodiments of the present invention also solve transmit sequence
35 number ambiguity by allowing some new and retransmit data frames to be sent with a variable number of sequence number bits. Described below are several methods by which the sender can determine the appropriate number of bits to use in the sequence number for new-data frames and retransmit data frames.

BRIEF DESCRIPTION OF THE DRAWINGS

The features, objects, and advantages of the present invention will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout and wherein:

FIG. 1 is a diagram of sequence number space and its use in several new RLP frames in accordance with an embodiment of the invention.

FIG. 2 is a diagram of sequence number space and its use in several RLP retransmit frames in accordance with an embodiment of the invention.

FIG. 3 is a flow chart of the steps used to transmit a data frame in accordance with an embodiment of the invention.

FIG. 4 is a flow chart of the steps used to extract data from a received data frame in accordance with an embodiment of the invention.

FIG. 5 is a diagram of a data communications system configured in accordance with an embodiment of the invention.

FIG. 6 is a flow chart of the steps used by a data receiver in an RLP link in sending NAK frames to the sender in accordance with an embodiment of the invention.

FIG. 7 is a flow chart of the steps used by a data sender in an RLP link in responding to received negative acknowledgment (NAK) frames and sending RLP retransmit frames in accordance with an embodiment of the invention.

FIG. 8 is a flow chart of the steps used by a data sender in an RLP link in choosing the number of bits in the sequence number of new-data frames in accordance with an embodiment of the invention.

FIG. 9 is a flow chart of the steps used by a data receiver in an RLP link in sending ACK frames to the sender in accordance with an embodiment of the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

FIG. 1 shows how sequence number space is used in an embodiment of the invention. New data 100 to be transmitted is placed into full-rate RLP frame 140. Within RLP frame 140 are a sequence number 150 and the data 100. New data 102 to be transmitted is placed into full-rate RLP frame 142. Within RLP frame 142 are a sequence number 152 and the data 102.

If new data 104 has fewer bytes than the maximum for a full-rate RLP frame, it is placed into non-full-rate RLP frame 144. Within RLP frame 144 are not only a sequence number 154 and the data 104, but also a data length 164.

As with RLP2, this embodiment of the invention allows the use of lower-rate frames within a given channel capacity, for example half-rate frames in a 9600bps channel, to carry smaller RLP frames.

In an exemplary embodiment of the invention, each sequence number
5 corresponds to the first byte of the data in the RLP frame. The sequence number carried within an RLP frame is called the RLP sequence number.

The data 100, 102, and 104 transmitted in the three RLP frames 140, 142, and 144, reside within sequence number space 110. In an exemplary embodiment of the invention, the sequence numbers are byte sequence
10 numbers, proceeding sequentially from low values on the left to high values on the right. In the current embodiment, as sequence numbers are either used or skipped, they increase monotonically. When a predetermined sequence maximum sequence number value is reached, the sequence numbers begin again at zero.

As the data bytes 102 placed into RLP frame 142, they are each assigned
15 sequential byte numbers from a block of sequence numbers 126. In an exemplary embodiment, the first byte of block 126 is a predetermined numeric distance 134a from the first byte of the block 122 used to transmit the previous RLP frame. This predetermined distance is called a page size, and sequence
20 numbers within the predetermined distance are collectively called a page. The sequence number block used to send data always starts at the beginning of a page. For example, sequence number block 122 starts at the beginning of page 134a, sequence number block 126 starts at the beginning of page 134b, sequence number block 130 starts at the beginning of page 134c. One side-effect of always
25 starting a new frame at the beginning of a page is that a block of sequence numbers at the end of a page, for example 124, 128, and 132, may not be assigned to transmitted data bytes.

In an exemplary embodiment of the invention, a shortened RLP
30 sequence number is used, which is equal to the byte sequence number of the first data byte in the RLP frame divided by the page size. The number of bits required to represent a shortened RLP sequence number is fewer than the bits required to represent the actual byte sequence number.

In another embodiment of the invention, the most-significant bits are
35 also omitted from the byte sequence number when such omission causes no ambiguity about which data is contained in the data portion of the RLP frame. For example, if a byte sequence number of 20 bits is used, but fewer than 2^{16} bytes are outstanding (have not been explicitly or implicitly acknowledged), the most significant 4 bits of the byte sequence number need not be sent in the RLP sequence number. These 4 most significant bits can be safely omitted from the

RLP sequence number without causing ambiguity in the sequence numbers of the bytes contained in the frame.

Though the shortening of the sequence number is portrayed herein as omitting a number of most significant bits from the sequence number, one skilled in the art will appreciate that the same result may be obtained by performing a modulo function on the sequence number without departing from the current invention.

If, however, more than 2^{16} bytes are outstanding, it may be that more than one of the outstanding data bytes have the same shortened RLP sequence number. In an exemplary embodiment of the invention, retransmit frames carrying such data bytes will include the entire 20-bit sequence number.

When an RLP frame is negatively acknowledged (NAK'd) and must be retransmitted, the data is inserted into RLP retransmit frames and retransmitted. If the channel capacity during retransmission is sufficient, then the retransmit frame is the same size as the original, lost RLP frame. In the case where the retransmit frame and the original RLP frame are the same size, the retransmit frame may use the same shortened RLP sequence number as the original, as long as doing so causes no sequence number ambiguity.

If an original transmit frame carries more data than will fit into a single retransmit frame, as is possible when channel capacity decreases, then the data from the original transmit frame is divided among several smaller retransmit frames. Each retransmit frame contains its own RLP sequence number, which may or may not be shortened in one of the ways previously discussed. If a retransmit frame is lost during transmission, that individual retransmit frame may be NAK'd, and subsequently retransmitted. In the rare event that channel capacity decreases before a retransmit frame is NAK'd, the data in that retransmit frame may be further divided among even smaller, independent retransmit frames before their second retransmission.

This procedure differs from RLP2, in which retransmit segments corresponding to a single lost frame all carry the same RLP sequence number. In order to recover the data in a single, lost retransmit segment, all retransmit segments carrying the sequence number of the lost frame must again be retransmitted.

FIG 2. shows the use of byte sequence number space and RLP sequence numbers when the data in a lost RLP frame must be retransmitted in several smaller RLP retransmit frames, and in accordance with an exemplary embodiment of the invention. The division of retransmit bytes may be necessary when channel capacity is reduced between the time of transmission of the original RLP frame and the retransmission of its data.

In the example shown, the sequence number space 228 used to represent the bytes in the original, lost RLP frame is divided among several RLP retransmit frames 230. Sequence number space 228 is divided into smaller portions 220, each having a size less than or equal to the capacity of full-rate frames on the current transmit channel. The data 200 corresponding to each of the smaller sequence number space portions 220 is placed into an RLP retransmit frame 230.

Each RLP retransmit segment has an RLP sequence number 240 corresponding to the first byte of its respective sequence number space 220. For example, sequence number 240_a has a value representative of the first value in sequence number space 220_a. The RLP sequence number 240 in each retransmit frame 230 may optionally be shortened in the same ways as discussed for RLP sequence numbers as long as doing so causes no sequence number ambiguity.

Each RLP retransmit frame may optionally have a data length 250. The data length 250 carried by each RLP retransmit frame indicates the number of data bytes 200 within the frame. For example, data length 250_a is equal to the number of data bytes 200_a in retransmit frame 230_a. If the length of data is indicated in other parts of the retransmit frame 230, for example in a type field not shown in the figure, the data length field 250 may be omitted.

Formation of RLP retransmit frames continues until the last portion of data 200_n is placed into an RLP retransmit frame 230_n. The last in a series of RLP retransmit frames often contains fewer than the maximum number of data bytes 200_n, and so typically contains a data length 250_n.

In the preferred embodiment of the invention, RLP sequence numbers for the most common transmit frames are shortened by omitting least significant bits and most significant bits. In an exemplary embodiment of the invention, the byte sequence number has 20 bits, a page size of 64 bytes is used, and the number of outstanding RLP transmit frames rarely exceeds 256.

Table 1. Example RLP Frame Headers

RLP Frame Header Fields								Frame Type Descriptions
Type bits	Bit0- Bit1	Bit2	Bit3	Bit4- Bit7	Byte 1	Byte 2	Byte 3	
11	8 bit seq#				Data			1. New: 8-bit RLP sequence number
10	8 bit seq#				Data			2. Retransmit: 8-bit RLP sequence number

01	8 bit seq#				Len	Data	3. New: 8-bit RLP sequence number, 8-bit length	
00	11	14bit seq#				Data	4. 14-bit RLP sequence number, No length	
00	10	14bit seq#				Len	Data	5. 14-bit RLP sequence number, 8-bit length
00	01	14bit seq#				Length		6. 14-bit RLP sequence number 16-bit length
00	00	1	1	20bit seq #			Len	7. Retransmit: Move to next page boundary.
00	00	1	0	20bit seq#			Len	8. Retransmit: Don't move to next page boundary
00	00	0	More formats for NAKs, Status, SYNC, ACK, SYNC/ACK, Idle, End of Frame					9. Extended control/sequence

Table 1 shows the fields of RLP frame headers used in accordance with the preferred embodiment of the invention. The contents of the frame headers may be placed at the beginning, the end, or dispersed deterministically throughout the RLP frame. In the preferred embodiment of the invention, the RLP frame header appears at the beginning of each RLP frame. Dispersion of the contents of the header throughout the frame may be desirable if a non-uniform probability of bit error exists throughout the RLP frame when received by the receiver.

A varying number of type bits are used to denote RLP frame type. Collectively, these type bits form a variable-bit-length type field, which indicates the type of RLP frame being transmitted, as well as the format of the rest of the RLP header. In **Table 1**, the "Frame Type Descriptions" column contains a description of the frame type corresponding to each type field value. The frame type descriptions in the "Frame Type Descriptions" column are identified with frame type numbers for purposes of discussion. All columns except the "Frame Type Descriptions" column show fields actually contained in the RLP header.

The type field is followed by the RLP sequence number. When possible without causing sequence number ambiguity, shortened RLP sequence

numbers of 8 bits are used. At other times, shortened RLP sequence numbers of 14 bits or full 20-bit RLP sequence numbers are contained by the RLP header.

Shortened RLP sequence numbers of 8 bits are generated from the 20-bit byte sequence number corresponding to the first data byte in the frame. A 20-bit byte sequence number is converted to an 8-bit shortened RLP sequence number by omitting the least significant 6 bits and the most significant 6 bits of the byte sequence number. The use of an 8-bit RLP sequence number is particularly advantageous, as 8-bit numbers are more easily manipulated by most modern microprocessors and in modern software. In the preferred embodiment of the invention, the enhanced RLP protocol is designed to maximize the number of RLP frames transmitted with 8-bit RLP sequence numbers.

A 20-bit byte sequence number is converted to an 14-bit shortened RLP sequence number by omitting the least significant 6 bits of the byte sequence number. When a 20-bit RLP sequence number is to be sent in an RLP frame, the 20-bit byte sequence number is simply copied into the RLP sequence number.

In the preferred embodiment of the invention, most frames carrying data to be transmitted for the first time use an 8-bit RLP sequence number. These frame types are identified in the "Frame Type Descriptions" column as types 1 and 3. In addition, RLP retransmit frames may also use 8-bit RLP sequence numbers, as indicated by type 2.

Frame types 4, 5, and 6 contain 14-bit RLP sequence numbers. These frames may optionally be used to avoid sequence number ambiguity in the event that an outstanding frame has the same 8-bit RLP sequence number as would otherwise be used by the next frame to be transmitted. These frames may also be optionally used in retransmitting data from negatively-acknowledged (NAK'd) frames.

Frame types 7 and 8 contain full 20-bit RLP sequence numbers. These frames are used when channel capacity decreases before NAK'd data can be retransmitted in fewer retransmit frames.

Here is an example of the use of retransmit frame types when channel capacity is reduced before retransmission. 750 bytes of data, bearing byte sequence numbers 1000 to 1749, are lost during transmission of a single RLP transmit frame on a 307kbps channel. By the time the receiver sends a NAK frame back to the transmitter, the channel capacity has been reduced to 9.6kbps. The 750 bytes of lost data must now be retransmitted over a 9.6kbps channel. In using FIG. 2 to represent this scenario, sequence number space 228 is the range of sequence numbers from 1000 to 1749. In a simple extension of RLP2, each retransmit frame using a frame type 8 header on a 9.6kbps channel carries 15

bytes of data. Using this as an example, the 750 bytes of retransmit data would be distributed among 50 RLP retransmit frames. In FIG. 2, RLP retransmit frame 230_n would have a frame type 8 header with a 20-bit sequence number of 1000 and a length of 15. With n=50, Frame 230_{n+1} would have a frame type 8 header with a 20-bit sequence number of 1720 and a length of 15. The use of frame type 8 indicates to the receiver that there are more data bytes assigned to the sequence number space between the retransmit frame and the next page boundary. Frame 230_n would have a frame type 7 header with a 20-bit sequence number of 1735 and a length of 15. The use of frame type 7 in the last retransmit frame indicates to the receiver that there is no data assigned to the sequence number space between the last byte in the frame and the next page boundary. Any frames of type 7 or 8 may be individually NAK'd and again retransmitted. In a simple improvement of the previous example, a variation of frame type 8 has a header with no length byte, allowing the frame to carry 16 bytes of data.

Frame type 9 carries the control/sequence frames used for synchronization in an RLP protocol. The use of control/sequence frames is described in detail in aforementioned RLP2.

In frame headers having no length field, the data length is interpreted to be the maximum number of bytes which can fit into the remainder of the transmitted frame. In this way, the enhanced RLP protocol described herein is easily extensible for use in channels having greater capacity than specified in aforementioned RLP2 or cdma2000. In many anticipated extensions, no modification of the enhanced RLP protocol would be necessary to accommodate new channel capacities.

The receiver, upon receiving RLP frames in the formats described herein, use shortened sequence numbers to reproduce the original byte sequence numbers to be applied to the bytes contained in each RLP frame. When a frame is received whose RLP sequence number indicates the loss of one or more preceding frames, the receiver sends a NAK to the transmitter. The NAK frame may optionally specify the number of bits to be used in the RLP sequence number of the corresponding RLP retransmit frame.

Though the preferred embodiment of the invention specifies the use of 20-bit byte sequence numbers and 8-, 14-, and 20-bit RLP sequence numbers, many other choices of sequence number sizes may be made without departing from the current invention. The use of RLP sequence numbers which enable unambiguous omission of most significant or least significant portions of the byte sequence number space are objects of various embodiments of the current invention, as are the selection of page sizes which are powers of 2 and the use of 8-bit shortened RLP sequence numbers.

FIG. 3 is a flow chart of the steps used to transmit a data frame in accordance with an embodiment of the invention. In the start of frame transmission processing 302, the transmitter evaluates 304 whether the transmitter must transmit a new RLP transmit frame or an RLP retransmit frame. This decision is based on whether NAK frames specifying lost transmitted data have been received previously.

Before forming a transmit frame, a transmit data buffer or queue is inspected to determine 306 whether the transmitter has new data to send. If there is no data which needs to be sent, an idle frame is formed 308 and transmitted 326. The use of SYNC, ACK, and IDLE frames is detailed in aforementioned RLP2. In one embodiment of the invention, idle frames are placed in smaller, lower-data-rate over-the-air frames to minimize the air link capacity impact of the idle frame transmission. In alternative embodiments of the invention, idle frames may not be transmitted at all, or are sent less frequently than every frame period.

If it is determined 306 that the transmitter has new data to send, the transmitter then evaluates 308 whether the data may be in a frame having an 8-bit RLP sequence number without causing sequence number ambiguity. If an 8-bit sequence number causes no ambiguity, then an RLP transmit frame having an 8-bit RLP sequence number is formed 314 and transmitted 326. If the number of data bytes to be sent is less than the maximum for a full-rate frame, the frame formed 314 will have a length field. If the number of data bytes to be sent is greater than or equal to the maximum for a full-rate frame, then that maximum number of data bytes is inserted into a full-rate frame 314, and that frame is transmitted 326. The RLP sequence number inserted into new transmit frames at steps 312 or 314 is generated from the byte sequence number beginning the next unused page.

If the transmitter determines 304 that the transmitter must transmit an RLP retransmit frame, then the next step is to determine 322 whether a full 20-bit RLP sequence number is required in the next RLP retransmit frame. A full 20-bit RLP sequence number is required if the data to be retransmitted must be retransmitted in more RLP frames than were used to originally transmit the data, as is possible when channel capacity is decreased in-between transmission and retransmission. If the next frame must be sent with a 20-bit RLP sequence number, then an RLP retransmit frame is formed and filled with data 324, and transmitted 326 with an RLP sequence number corresponding to the first byte of data in the RLP retransmit frame. If the transmitter determines 322 that a 20-bit RLP sequence number is not necessary, then the transmitter must then choose 318 between the use of an 8-bit or a 14-bit RLP sequence number for the

next retransmitted RLP frame. Once the determination of required sequence number bit-size is complete 318, the transmitter forms an RLP transmit frame with an 8-bit RLP sequence number 320, or forms an RLP transmit frame with a 14-bit RLP sequence number 316 and transmits the frame 326. In either case 316 or 320, the RLP sequence number inserted into the frame will be generated from the byte sequence number corresponding to the first byte of the data contained in the frame. That byte sequence number will be the first byte sequence number of a sequence number page. If necessary to avoid ambiguity about data size, the RLP header of the retransmitted RLP frame will contain a length field.

FIG. 4 is a flow chart of the steps used to extract data from a received data frame in accordance with an embodiment of the invention. Upon receiving an RLP data frame or an idle frame containing an RLP sequence number 402, the receiver expands the received RLP sequence number from the RLP frame as necessary to form a byte sequence number, and extracts any data bytes from the frame 404. The receiver then determines 406, from that byte sequence number whether RLP frames carrying data have been lost.

In an exemplary embodiment of the invention, the determination of lost data is made by comparing the byte sequence number indicated by the most recently received RLP data frame to the byte sequence number corresponding to the byte sequence number of the latest previous data byte successfully received. For example, the last byte of a received RLP data frame which is not a retransmission is considered the latest previous data byte successfully received. If the byte sequence number expanded from a received RLP data frame which is not a retransmission, or of a received RLP idle frame, is different by greater than a page from the sequence number of the latest previous data byte successfully received, then the loss of one or more lost RLP data frames has occurred.

The aforementioned RLP2 protocol describes the use of timers and frame counters in determining the loss of retransmitted RLP frames. These techniques are also used in embodiments of the current invention.

Upon determining that data has been lost 406, the receiver forms and sends a NAK frame 408 to request retransmission of the lost data. In an alternate embodiment of the invention, the receiver may specify, within the NAK frame, a reduced number of bits to be used in the RLP sequence numbers within retransmitted RLP frames, based on the receiver's knowledge of byte sequence number space of bytes successfully received.

Upon determining that data has not been lost 406, the receiver forms and sends RLP idle or RLP data frames as appropriate. In alternative embodiments

of the invention, idle frames may not be transmitted at all, or are sent less frequently than every frame period.

FIG. 5 is a diagram of a data communications system configured in accordance with an embodiment of the invention. As shown, transmitter 502 communicates with receiver 504 over a wireless communication channel 506. RLP data frames and control frames are sent from transmitter 502 to receiver 504 along channel 506a, and RLP acknowledgments or NAK's are sent from receiver 504 to transmitter 502 along channel 506b. During transmission, the capacity of transmission channel 506 may change, sometimes requiring retransmission of data over a channel of lesser capacity than used in the original transmission.

For example, a large, full-rate RLP data frame may be sent by transmitter 502 over channel 506a, but not received successfully by receiver 504. Then, the capacity of channel 506 is reduced either shortly before or shortly after receiver 504 sends a NAK frame for the lost data on channel 506b. Transmitter 502 must now distribute the data bytes from the original lost full-rate RLP data frame among several smaller RLP frames for retransmission over reduced-capacity channel 506b. If receiver 504 determines that one or more retransmitted RLP frames have been lost upon retransmission over reduced-capacity channel 506a, receiver 504 may individually NAK those individual lost retransmit frames over channel 506b.

In an exemplary embodiment of the invention, transmitter 502 includes a processor connected to a memory and an apparatus for wirelessly transmitting and receiving bytes processed by the processor. The processor is provided with a byte stream to be transmitted over the transmit apparatus and forms transmissions in accordance with the enhanced radio link protocol (RLP) described hereinbefore.

In an exemplary embodiment of the invention, receiver 504 includes a processor connected to a memory and an apparatus for wirelessly transmitting and receiving bytes processed by the processor. The processor is provided with received data received from the receive apparatus and forms response frames for transmission in accordance with the enhanced radio link protocol (RLP) described hereinbefore.

Various embodiments of the present invention used to prevent sequence number ambiguity are discussed in relation to **FIG. 6** through **FIG. 9**. Some of the described embodiments use different numbers of bits to represent full and shortened RLP sequence numbers. To avoid confusion when discussing sequence number variables introduced in RLP2, such as L_V(R), L_V(N), L_V(S) and their shortened versions V(R), V(N), V(S), this application simply refers to

them as $V(R)$, $V(N)$ and $V(S)$, and supplements them where necessary with the words "full" and "shortened".

Also, because sequence numbers are represented by a fixed number of bits, their values can "wrap around," causing a subsequent sequence number to sometimes have an arithmetically lesser value than an earlier one. In comparing two sequence numbers, it is said that a first sequence number is "less than" a second sequence number when it is used to carry data which precedes that of a frame having the second sequence number. The second sequence number regardless of arithmetic relationships, is said to be "greater than" the first sequence number.

FIG. 6 is a flow chart of the steps used by a data receiver in an RLP link in sending negative acknowledgment (NAK) frames to the sender in accordance with an embodiment of the invention. The flow chart begins at the stage 602 with the determination that the data receiver must send a NAK frame to the data sender. This determination is usually based on the receipt of a new-data frame having a full sequence number greater than $V(R)$, which indicates that one or more frames sent by the data sender were not received by the data receiver.

During evaluation step 604, the data receiver determines which of several types of retransmission frames to request. The difference between the several types of RLP retransmission frames is the number of bits used by the RLP sequence number field 'SEQ'. At first, the size of 'SEQ' might seem unimportant, because it represents such a small portion of most RLP data frames. In the case of RLP2, the 'SEQ' field is 8 bits. The difference of a few sequence number bits in a retransmit frame is significant, however, because these few extra bits can often necessitate segmentation of a retransmit frame into multiple frames. This is undesirable, because every additional retransmit frame incurs the cost of frame overhead, thus consuming extra bandwidth.

The majority of new-data frames sent over an RLP link are typically full-rate data frames. In the enhanced protocol described above, a requested retransmit frame will often have been originally sent in a frame format that utilized a shortened sequence number to allow squeezing the maximum number of data bytes into the frame. The frame formats used with full-rate frames in RLP are designed to use every possible bit in the frame for carrying the data payload. If 'SEQ' suddenly grows by even a single bit, it can cause the frame to be too small to carry the original data. As a result, the data must be segmented and divided among two RLP retransmit frames. For this reason, it is much more efficient to retransmit full-rate RLP frames using the same shortened sequence number as the original whenever possible.

In accordance with the present invention, sequence numbers may be converted to shortened sequence numbers in a variety of ways. Using the methods described in the aforementioned RLP2, 12-bit sequence numbers are shortened to 8 bits by performing a modulo 256 function on the full 12-bit *frame* sequence number. In an exemplary embodiment using *byte* sequence numbers, a 20-bit sequence number may be shortened to either 8 or 14 bits as described above for use with a dynamic capacity channel.

In an exemplary embodiment of the present invention, the NAK frame format includes a 'RexmitSeqSize' field specifying the size of the 'SEQ' field to be used in the requested retransmit frames. As in RLP2, NAK frame optionally contain 'FIRST' and 'LAST' fields, indicating a range of consecutive frames to be retransmitted, or bit maps identifying the sequence number values of non-consecutive data frames to be retransmitted. In an exemplary embodiment, the 'FIRST' and 'LAST' fields in a NAK frame are always represented by full-sized sequence numbers. In an exemplary embodiment where both sides of the link maintain 12-bit V(N), V(R) and V(S) *frame* sequence numbers, the 'FIRST' and 'LAST' fields in the NAK packet will also be 12 bits. In another exemplary embodiment where both sides of the link maintain 20-bit V(N), V(R) and V(S) *byte* sequence numbers, the 'FIRST' and 'LAST' fields in the NAK packet will both be 20 bits.

In an exemplary embodiment that utilizes two sizes of sequence numbers in retransmit frames, the 'RexmitSeqSize' field is represented by a single bit in the NAK frame. In an exemplary embodiment, a '1' value in the 'RexmitSeqSize' field is used to request a retransmit frame with a 12-bit sequence number, and a '0' is used to request a retransmit frame with a shortened 8-bit sequence number. In another exemplary embodiment that utilizes three sizes of sequence numbers in retransmit frames, the 'RexmitSeqSize' field is represented by two bits in the NAK frame. One skilled in the art will appreciate that the number of sequence number sizes used and the specific values chosen may be varied without departing from the present invention.

By specifying a number of bits to be used in retransmit frames, the data receiver controls the type of retransmission frames sent in response to the NAK frames. The frame size decision in step 604 is based on a desire to ask for the smallest sequence number size which will avoid sequence number ambiguity in retransmit frames. The 'RexmitSeqSize' field is specified such that the entire full RLP sequence number can be unambiguously reconstructed from the sequence number bits in the subsequent received retransmit frame.

In an exemplary embodiment, using 12-bit frame sequence numbers step 604 chooses a sequence number size of either 12 or 8 bits. In this embodiment, $V(N)$ and $V(R)$ are compared to each other according to equation (1).

$$(V(N) + 256 - V(R)) \bmod 256 \geq 256 \quad (1)$$

- 5 Using this equation, the size of the sequence number space between $V(N)$ and $V(R)$ is compared to the size of the sliding window covered by a sequence number of n bits (in this example, n is equal to 8). If the sequence number space is larger than 2^n , then a NAK is sent requesting that the full 12-bit sequence number ('SEQ' field) be used in corresponding retransmit frames 608.
- 10 Otherwise, a NAK is sent requesting that the shortened sequence number ('SEQ' field) be used in corresponding retransmit frames 606. Using equation (1) to determine the size of 'SEQ' is advantageous, because common microprocessors can perform the comparison quickly and efficiently.

- 15 In another embodiment, the data receiver compiles a "hole list" of all the sequence numbers between $V(N)$ and $V(R)$ which were not received. The sequence numbers in this hole list represent those which must be negatively acknowledged (NAK'd). Retransmit sequence number ambiguity can only exist if one shortened sequence number can be used to represent more than one full-length sequence number in the hole list. A set of more than one full-length
- 20 sequence number corresponding to the same shortened sequence number is called a "match."

- In an exemplary embodiment using a hole list, the size of sequence number requested is based, not on equation (1), but on the contents of the hole list. If a match is found among sequence numbers in the hole list, the use of
- 25 shortened sequence numbers in associated retransmit frames can potentially cause ambiguity for the data receiver.

- In an exemplary embodiment of the invention, the data receiver solves this problem by requesting that all retransmit frames having sequence numbers in a match be transmitted with full sequence numbers. In another embodiment,
- 30 the data receiver chooses one sequence number of the match for retransmission with a shortened sequence number, and specifies that the rest be retransmitted with full sequence numbers. When the retransmission arrives with a shortened sequence number, the data receiver can identify its full sequence number according to which NAK frames were sent requesting shortened sequence
- 35 numbers. In the preferred embodiment of the invention, the earliest sequence number of a match is chosen for retransmission using a shortened sequence number.

In another embodiment using a hole list, the data receiver only sends a NAK for the earlier sequence number of a match. The NAK for the latter sequence number of a match is not sent until a retransmit frame for the earlier sequence number is successfully received. In an exemplary embodiment, there is no need to request full sequence numbers in retransmit frames, because the data receiver uses the round-trip-delay of the link to identify the NAK corresponding to each received retransmit frame.

Depending on the implementation, searching through the hole list may require more computational power than using the equation (1), but it minimizes unnecessary requests for long sequence numbers. Using equation (1) can sometimes lead to requesting a long sequence number when no ambiguity would result from using a shortened sequence number.

After either sending a NAK requesting a shortened 'SEQ' 606 or sending a NAK requesting a full 'SEQ' 608, the method of FIG. 6 continues on to process other received frames 610.

FIG. 7 is a flow chart of the steps used by a data sender in an RLP link in responding to received NAK frames and sending RLP retransmit frames in accordance with an embodiment of the invention. Upon receiving a NAK frame 702, as described in conjunction with FIG. 6, the 'RexmitSeqSize' field is inspected 704 to determine the size of sequence number requested in retransmission. If the NAK frame allows the use of a shortened sequence number in the retransmit frame, then the corresponding retransmit frame or frames are sent 706 using shortened sequence numbers. If the NAK frame contains a request for full-sized sequence numbers, then the corresponding retransmit frame or frames are sent 708 using full-sized or "full" sequence numbers. As mentioned above, the use of full sequence numbers may, by itself, necessitate segmentation in retransmitting a full-rate frame. After sending the requested retransmit frames, the method of FIG. 7 then continues to process other frames 710.

FIG. 8 is a flow chart of the steps used by a data sender in a radio link protocol (RLP) link in choosing the number of bits in the sequence number of new-data frames in accordance with an embodiment of the invention. The exemplary embodiment of an RLP protocol shown in FIG. 8 is unusual in that the data sender keeps track of the value of $V(R)_R$ belonging to the data receiver. This is a departure from previous forms of RLP, in which the data sender tracks only its own variables, $V(N)_S$, $V(R)_S$ and $V(S)_S$. In this discussion, the references to the normal RLP variables $V(N)$, $V(R)$ and $V(S)$ include subscripts "R" to denote RLP variables belonging to the data receiver, and "S" to denote RLP variables belonging to the data sender.

The flow chart begins with the determination 802 that the data sender has new data to send in a new-data frame. An analysis is performed 804 to determine the size of the sequence number to use in the new-data frame. In the exemplary embodiment, the data sender uses the value of $V(R)_R$ to decide
 5 whether to send a full sequence number or a shortened sequence number in the new-data frame to be sent.

As discussed above, full sequence numbers may be converted to shortened sequence numbers in a variety of ways. Regardless of which conversion method is used, it is essential, that a new frame not be sent with a shortened sequence number if the corresponding full sequence number cannot
 10 be unambiguously identified by the data receiver. In RLP2, for example, both sides maintain 12-bit variables for $V(N)$, $V(R)$ and $V(S)$, but the sequence numbers are shortened to 8 bits using a modulo 256 function. The shortened 8-bit sequence numbers can be used to unambiguously identify sequence
 15 numbers from $V(R)$ to $V(R)+255$. Beyond this sequence number range, the use of an 8-bit sequence number is no longer effective. As discussed previously, high-data-rate channels make the exhaustion of this 8-bit sequence number range far more likely.

In the embodiment shown in FIG. 8, the choice between an 8-bit or a 12-bit sequence number size is based on a comparison of $V(R)_R$ and $V(S)_S$, as shown
 20 in equation (2).

$$V(S)_S \geq V(R)_R + 256 \quad (2)$$

In accordance with this equation, the size of the sequence number chosen for the new-data frame in step 804 will be 12 bits if $V(S)_S$ is greater than or equal to
 25 $V(R)_R+256$, and 8 bits otherwise. In a system using a different method of shortening sequence numbers, the conditional comparison of equation (2) is adjusted as necessary to avoid new-data frame sequence number ambiguity.

In an exemplary embodiment, in a system utilizing 8- and 14-bit shortened byte sequence numbers as described above, equation (2) would be
 30 unchanged, because the 8-bit sequence number is equivalent to a frame sequence number.

Following the determination of appropriate sequence number size 804, either a new-data frame with a shortened 'SEQ' is sent 806, or a new-data frame with a full-sized 'SEQ' is sent 808. Following the transmission of the
 35 appropriate new-data frame, the method of FIG. 8 continues on to process the next frame 810.

FIG. 9 is a flow chart of the steps used by a data receiver in an RLP link in sending STATUS or ACK control (hereafter referred to as ACK) frames to the sender in accordance with an embodiment of the invention. The flow chart begins with the determination 902 that it is time for the data receiver to send a frame to the data sender. At step 904, the data receiver determines whether there is a need to send an acknowledgment (ACK) frame. Since the radio link protocol (RLP) is generally a NAK-based protocol, the sending of an ACK frame over an RLP link is a departure from previous designs. In the embodiments of the present invention described below, the ACK frame contains the data receiver's current sequence number value of $V(R)_R$. In addition, the ACK frame optionally contains $V(N)_R$ to enable the data sender free up memory used by a send queue. Also, the ACK frame optionally contains $V(S)_R$ to enable the data sender to determine the need to send a NACK to the data receiver for data flowing in the opposite direction.

In exemplary embodiments, the decision 904 to send an ACK frame is based on several possible criteria. In a first embodiment, the receipt by the data receiver of a new-data frame having a full sequence number triggers the data receiver to send one or more ACK frames to update the data sender's copy of $V(R)_R$. In a second embodiment, the data receiver sends an ACK frame whenever the sequence numbers in received new-data frames exceeds the last transmitted $V(R)_R$ value sent by a predetermined value. For example, if the $V(R)_R$ value sent in the last ACK frame was X, a new ACK frame is sent whenever the 8-bit shortened sequence number of a new-data frame is received which exceeds $X+128$. In a third embodiment, ACK frames are sent periodically, or every time a predetermined number of frame periods has lapsed.

In a fourth embodiment, a combination of the first three methods is used to determine when to send the next ACK frame. For example, the receipt of a new-data frame having a full sequence number triggers the data receiver to send an ACK. In addition, the receipt of new-data frames having full sequence numbers is generally avoided by sending an ACK whenever the received shortened sequence number crosses some threshold beyond the previously-sent $V(R)_R$ value. Regardless of frames received, an ACK is additionally sent if no ACK has been sent for a predetermined number of previous consecutive frames (for example 200 frames).

If the decision is made 904 not to send an ACK frame, then a data or idle frame is sent from the data receiver to the data sender 912. One familiar with the art will appreciate that an RLP implementation may send no frame at all at step 912 without departing from the present invention.

If the decision is made 904 to send an ACK frame, then the availability of new data to send is evaluated at step 906. If there is new data to send, then a frame containing new data and the ACK information is sent 908. Otherwise, a plain ACK frame is sent 910. After an appropriate frame is sent (or not sent as discussed in conjunction with step 912), the method continues 914, eventually returning to step 902.

In an exemplary embodiment of the invention, the set of sequence number bit sizes and the use of frame sequence numbers or byte sequence numbers is negotiated by the data sender and the data receiver when a data call is first established. The characteristics of RLP protocol used between the data sender and the data receiver are chosen from several possible configurations. This negotiation lends even greater flexibility to the RLP enhancements of the present invention.

The embodiments of the present invention described form a family of complete solutions to efficiency problems posed by extending existing versions of RLP to high-data-rate and dynamic capacity channels. The embodiments described exhibit a variety of possible combinations resulting from analysis of the tradeoffs associated with each.

The previous description of the preferred embodiments is provided to enable any person skilled in the art to make or use the present invention. The various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without the use of the inventive faculty. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.